

In the Claims

1. **(Currently Amended)** A job scheduling device for scheduling jobs to run on at least two nodes of at least one computing platform, comprising:

at least two local job repositories, each local job repository installed on a separate one of the at least two nodes and each local job repository configured to maintain job information on each job submitted to the node where the local job repository is installed, the job information including job parameters needed to execute each job; and

at least two enterprise scheduling agents, each enterprise scheduling agent installed on a separate one of the at least two nodes and each enterprise scheduling agent configured to:

access the job information maintained by the local job repository;

schedule for execution each job submitted to the node where the local job repository is installed;

determine when to execute each job submitted to the node where the local job repository is installed; and

launch execution of each job submitted to the node based on the determination;

a presentation system configured to accept and validate parameters identifying at least one job to be submitted for execution on at least one of the nodes; and

a job scheduler configured to allocate at least one job based on the parameters to at least one of the nodes and to submit the allocated jobs to the at least one of the nodes;

wherein the job scheduler is communicatively coupled to the at least two nodes by a network.

2. **(Previously Presented)** The job scheduling device according to Claim 1, further comprising:

a job data management device configured to maintain job data and job histories of jobs submitted to each of the nodes.

3. **(Previously Presented)** The job scheduling device according to Claim 2, wherein the job histories include information received from each enterprise scheduling agent regarding status of the jobs submitted.

4. **(Previously Presented)** The job scheduling device according to Claim 2, wherein the job data management device is utilized by the job scheduler to set parameters in jobs to be submitted to each of the nodes.

5. **(Previously Presented)** The job scheduling device according to Claim 1, further comprising:

a job history repository that saves both jobs and job histories of jobs submitted to each enterprise scheduling agent;

wherein each enterprise scheduling agent comprises,

an agent communicator configured to send and receive messages between the job scheduler and the enterprise scheduling agent,

a job manager configured to setup, launch, run, and manage jobs submitted to the enterprise scheduling agent,

a data manager configured to update and delete data from the job history repository, and

a low level API configured to handle internal functions of the enterprise scheduling agent, file management, and message handling functions.

6. **(Previously Presented)** The job scheduling device according to Claim 5, further comprising:

an enterprise communicator configured to construct and communicate messages between the job scheduler and each enterprise scheduling agent; and

a job data management device configured to maintain job histories of jobs submitted to each enterprise scheduling agent;

wherein the data manager updates the job history via enterprise communicator messages sent from each enterprise scheduling agent to the job data management device.

7. **(Previously Presented)** The job scheduling device according to Claim 1, further comprising:

a command line device configured to accept commands regarding administration of jobs submitted to at least one of the nodes and

a job administration device configured to communicate the command line to the enterprise scheduling agent installed on the at least one of the nodes for execution.

8. **(Previously Presented)** The job scheduling device according to Claim 7, wherein:

the commands accepted by the command line device include at least one of delete a job and all runs of the job, cancel a job's run, list all jobs by at least one of product code, status, and node, and rerun a job immediately.

9. **(Previously Presented)** The job scheduling device according to Claim 8, wherein:

the commands accepted by the command line device include context variables; and

each enterprise scheduling agent converts the context variables according to a current job and job parameters, and executes the commands.

10. **(Previously Presented)** The job scheduling device according to Claim 1, further comprising:

a point product device configured to provide a communication link between each enterprise scheduling agent and at least one product submitting jobs to the job scheduling device;

wherein the point product device communicates job status, job logfile, setup, cancel, job parameter functions, and requests between each enterprise scheduling agent and the at least one product.

11. **(Previously Presented)** The job scheduling device according to Claim 10, further comprising:

a job administration device configured to accept command line inputs and communicate the command line inputs to at least one enterprise scheduling agent;

a job data management device configured to maintain job histories of jobs submitted to each of the nodes; and

an enterprise communicator configured to send messages between at least one of said job scheduler, point product device, job administration device, and job data management device and each enterprise scheduling agent.

12. **(Previously Presented)** The job scheduling device according to Claim 1, further comprising:

an enterprise communicator configured to send messages between the job scheduler and each enterprise scheduling agent.

13. **(Previously Presented)** The job scheduling device according to Claim 12, wherein:

each enterprise scheduling agent is registered at a specific node address that identifies each enterprise scheduling agent with a unique datagroup; and

the enterprise communicator encodes each message with at least one destination corresponding to a datagroup to direct each message to at least one enterprise scheduling agent.

14. **(Previously Presented)** The job scheduling device according to Claim 1, wherein:

each local job repository maintains job history information on each job submitted to the node where the local job repository is installed; and

each local job repository is updated by the enterprise scheduling agent installed on the node where the local job repository is installed.

15. **(Previously Presented)** The job scheduling device according to Claim 14, further comprising:

a job data management device configured to maintain job histories of jobs submitted to each of the nodes; and

a synchronizing device configured to synchronize each local job repository with the job histories maintained by the job data management device.

16. **(Previously Presented)** The job scheduling device according to Claim 1, further comprising:

a progress monitor configured to monitor and display execution of at least one of the jobs;

wherein:

the progress monitor provides a visual display of,
an identification of the job and a current phase of the job,
a percentage complete of the job, and
a percentage complete of the current phase.

17. **(Previously Presented)** The job scheduling device according to Claim 1, further comprising:

an auto login device configured to accept login parameters from a user submitting a job;

wherein the login parameters are utilized by each enterprise scheduling agent to launch and execute the job submitted.

18. **(Previously Presented)** The job scheduling device according to Claim 1, further comprising:

a notification scripting device configured to execute a notification script having instructions for notifying a user of status or a submitted job;

wherein the notification scripting device includes facilities for creating, editing, and selecting a notification script for a specific job.

19. **(Previously Presented)** The job scheduling device according to Claim 1, wherein:

the presentation system includes,

a GUI interface that accepts user inputs for scheduling and specifying a job to be submitted;

wherein the GUI interface includes facilities for selection and creation of a scheduling calendar, selection of a start date and time, selection of recurring job run intervals, and selection of an immediate job run.

20. **(Original)** The job scheduling device according to Claim 1, further comprising:

a resource management device configured to enable a user to locate and view jobs and job runs.

21. **(Previously Presented)** The job scheduling device according to Claim 20, wherein:

the resource management device includes a GUI for defining an object representing a job, having,

a general properties page having input fields for a label identifying the job, and a description of the job,

a description properties page having a selection field for identifying an icon for representing the job, and

a repository page having a selection field for identifying a time zone for display of job times.

22. **(Previously Presented)** The job scheduling device according to Claim 21, wherein:

objects defined by the resource management device comprise,

a hierarchy of folders including at least one of an all jobs folder, a jobs by group folder, a jobs by node folder, a jobs by product folder, a jobs by type folder, and a jobs by user folder.

23. **(Previously Presented)** The job scheduling device according to Claim 22, wherein the all jobs folder includes folders, including,

an all jobs any status folder listing jobs regardless of status and associated job history of each job,

an all runs by status folder listing jobs according to status, including completed runs, failed runs, not started runs, preempted runs, running runs, and stopped runs,

a held jobs folder listing jobs that are held and can be scheduled for a later time, and

a scheduled jobs folder listing jobs that are scheduled to run.

24. **(Previously Presented)** The job scheduling device according to Claim 1, wherein:

the presentation system includes, a strategy scheduling window configured to allow a user to view, create, modify, and delete schedules for a strategy.

25. **(Currently Amended)** A method of scheduling jobs to run on at least two nodes of at least one computing platform, comprising:

determining, at a first location, at least one job to be scheduled based on job parameters for the at least one job;

submitting the at least one job to the at least two nodes, each node having a local job repository and an enterprise scheduling agent installed thereon;

maintaining, at each of the local job repositories, the at least one job submitted to the node where the local job repository is installed;

using each scheduling agent to schedule for execution each job submitted to the node where the local job repository is installed;

determining, at each of the enterprise scheduling agents, when to execute the at least one job submitted to the node where the enterprise scheduling agent is installed; and

based on the determinations, executing the at least one job on the at least two nodes under management of the enterprise scheduling agents;

wherein the first location is communicatively coupled to the at least two nodes by a network.

26. **(Previously Presented)** The method according to Claim 25, further comprising:

monitoring progress of each job executing on the at least two nodes; and
displaying the progress on a progress monitor.

27. **(Previously Presented)** The method according to Claim 25, further comprising recording each job and a history of each job in a job history repository.

28. **(Previously Presented)** The method according to Claim 27, further comprising:

utilizing a job data management device for,
retrieving status messages regarding each job sent from each enterprise scheduling agent, and
updating said job history repository based on said status messages.

29. **(Previously Presented)** The method according to Claim 28, further comprising:

maintaining, at each of the local job repositories, job history information for each job submitted to the node where the local job repository is installed.

30. **(Previously Presented)** The method according to Claim 29, further comprising:

synchronizing the job history repository with each local job repository.

31. **(Previously Presented)** The method according to Claim 25, wherein determining at least one job to be scheduled comprises:

retrieving the job parameters from one of a product and a user interface that collects the job parameters;
validating the job parameters; and
allocating a job based on the job parameters.

32. **(Previously Presented)** The method according to Claim 25, wherein submitting the at least one job comprises:

packaging the job parameters in a communication format; and
transmitting the packaged job parameters from a computing platform where the job parameters are determined to the at least two nodes.

33. **(Previously Presented)** The method according to Claim 25, wherein executing the at least one job comprises:

setting up the at least two nodes to run an application program identified by the job parameters;
executing the application program on the at least two nodes; and
monitoring progress of the application being executed on the at least two nodes.

34. **(Previously Presented)** The method according to Claim 25, further comprising:

accepting a command line for administration of a job submitted to at least one of the enterprise scheduling agents; and
communicating the command line to the at least one of the enterprise scheduling agents for execution.

35. **(Previously Presented)** The method according to Claim 34, further comprising:

substituting context variables in the command line with data based on the context variable and the job to be administered; and
executing the command line.

36. **(Previously Presented)** The method according to Claim 25, further comprising:

communicating data, including at least one of job status, job logfile, setup, cancel, job parameter functions, and requests for the data between a product and each enterprise scheduling agent.

37. **(Previously Presented)** The method according to Claim 25, further comprising:

registering each enterprise scheduling agent at a node address that identifies the registered enterprise scheduling agent with a unique datagroup;

communicating jobs and job administration commands and requests with each enterprise scheduling agent via messages; and

encoding each message sent to a recipient enterprise scheduling agent with at least one destination corresponding to a datagroup that directs the message to the recipient enterprise scheduling agent.

38. **(Previously Presented)** The method according to Claim 25, further comprising:

retrieving auto login parameters from a user scheduling an auto login job; and

launching execution of the job utilizing the auto login parameters.

39. **(Previously Presented)** The method according to Claim 38, further comprising:

retrieving a notification script for a job being submitted; and

executing the notification script on at least one of completion of the job and at a requested status point.

40. **(Previously Presented)** The method according to Claim 25, further comprising:

accepting a scheduling calendar identifying at least one of an execution time and an interval for at least one of the jobs; and

executing the jobs on the at least two nodes at the time and interval identified in the calendar.

41. **(Previously Presented)** The method according to Claim 25, further comprising:

providing a description of at least one of the jobs, including a written description, a label, and an icon selected to represent the job; and

identifying a time zone for display of job times.

42. **(Previously Presented)** The method according to Claim 25, further comprising:

placing information about job times and status in an object containing folders, each folder identifying a categorization of jobs contained therein, including, an all jobs folder, a jobs by group folder, a jobs by node folder, a jobs by product folder, a jobs by type folder, and a jobs by user folder.

43. **(Previously Presented)** The method according to Claim 42, further comprising:

organizing the all jobs folder to maintain additional folders, including, at least one of, an all jobs any status folder listing jobs regardless of status and associated job history of each job,

an all runs by status folder listing jobs according to status, including completed runs, failed runs, not started runs, preempted runs, running runs, and stopped runs,

a held jobs folder listing jobs that are held and can be scheduled for a later time, and

a scheduled jobs folder listing jobs that are scheduled to run.

44. **(Previously Presented)** The method according to Claim 25, further comprising providing a strategy scheduling window that allows a user to view, create, modify, and delete schedules for a strategy.

45. **(Currently Amended)** Software for use in scheduling jobs to run on at least two nodes of at least one computing platform, the software embodied in computer readable media that, when executed using one or more computers, is operable to:

determine, at a first location, at least one job to be scheduled based on job parameters for the at least one job;

submit the at least one job to the at least two nodes, each node having a local job repository and an enterprise scheduling agent installed thereon;

maintain, at each of the local job repositories, the at least one job submitted to the node where the local job repository is installed;

schedule for execution each job submitted to the node where the local job repository is installed;

determine, at each of the enterprise scheduling agents, when to execute the at least one job submitted to the node where the enterprise scheduling agent is installed; and

based on the determinations, execute the at least one job on the at least two nodes under management of the enterprise scheduling agents;

wherein the first location is communicatively coupled to the at least two nodes by a network.

46. **(Currently Amended)** A job scheduling device for scheduling jobs to run on at least two nodes of at least one computing platform, comprising:

installed on each of the at least two nodes:

first means for maintaining job information on each job submitted to the node where the first means is installed, the job information including job parameters needed to execute the job;

second means for:

accessing the job information maintained by the first means;

scheduling for execution each job submitted to the node where the first means is installed;

determining when to execute the jobs submitted to the node where the first means is installed; and

launching execution of each job submitted to the node based on the determination;

third means for accepting and validating parameters identifying at least one job to be submitted for execution on at least one of the nodes; and

fourth means for allocating at least one job based on the parameters to at least one of the nodes and for submitting the allocated jobs to the at least one of the nodes;

wherein the first means is communicatively coupled to the at least two nodes by a network.

47. **(Previously Presented)** The job scheduling device as recited in Claim 1, wherein the presentation system comprises a Graphic User Interface Application Program Interface (API GUI).

48. **(Previously Presented)** The job scheduling device as recited in Claim 46, wherein the third means comprises a Graphic User Interface Application Program Interface (API GUI).

49. **(Currently Amended)** A job scheduling system for scheduling jobs to run on at least two nodes of at least one computing platform, comprising:

at least two local job repositories, each local job repository installed on a separate one of the at least two nodes and each local job repository configured to maintain job information on each job submitted to the node where the local job repository is installed, the job information including job parameters needed to execute each job; and

at least two enterprise scheduling agents, each enterprise scheduling agent installed on a separate one of the at least two nodes and each enterprise scheduling agent configured to:

access the job information maintained by the local job repository;

schedule for execution each job submitted to the node where the local job repository is installed;

determine when to execute the jobs submitted to the node where the local job repository is installed; and

launch execution of each job submitted to the node based on the determination;

a presentation system configured to accept and validate parameters identifying at least one job to be submitted for execution on at least one of the nodes; and

a job scheduler configured to allocate at least one job based on the parameters to at least one of the nodes and to submit the allocated jobs to the at least one of the nodes so that the at least one allocated job can be executed by the enterprise scheduling agent installed on the node to which the job was allocated and submitted;

wherein the job scheduler is communicatively coupled to the at least two nodes by a network

50. **(Currently Amended)** A job scheduling method for scheduling jobs to run on at least two nodes of at least one computing platform, comprising:

installing at least two local job repositories, each local job repository installed on a separate one of the at least two nodes, each local job repository configured to maintain job information on each job submitted to the node where the local job repository is installed, the job information including job parameters needed to execute each job;

installing at least two enterprise scheduling agents, each enterprise scheduling agent installed on a separate one of the at least two nodes, each enterprise scheduling agent configured to:

access the job information maintained by the local job repository;

schedule for execution each job submitted to the node where the local job repository is installed;

determine when to execute the each job submitted to the node where the local job repository is installed; and

launch execution of each job submitted to the node based on the determination;

at a first location:

accepting and validating parameters identifying at least one job to be submitted for execution on at least one of the nodes; and

allocating at least one job based on the parameters to at least one of the nodes;

submitting the at least one allocated job to the at least one of the nodes; and

executing the at least one allocated job on the at least one of the nodes to which the job was allocated and submitted;

wherein the first location is communicatively coupled to the at least two nodes by a network.